<table>
<tr><th>L Number</th><th>Hits</th><th>Search Text</th><th>DB</th><th>Time stamp</th></tr>
<tr><td>1</td><td>14</td><td>((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/23 09:54</td></tr>
</table>

2	14	(((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)	USPAT; US-PGPUB; EPO; JPO	2003/08/23 10:00

<table>
<tr><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td>3</td><td>6</td><td>((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5</td><td>USPAT;
US-PGPUB;
EPO; JPO</td><td>2003/08/23
10:08</td></tr>
</table>

<table>
<tr><td>4</td><td>6</td><td>(((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and (ceas$5 or stop$5)</td><td>USPAT;
US-PGPUB;
EPO; JPO</td><td>2003/08/23
10:08</td></tr>
</table>

<table>
<tr><td>5</td><td>6</td><td>(((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and (ceas$5 or stop$5) and provid$5</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/23 10:53</td></tr>
</table>

<table>
<tr><td>6</td><td>0</td><td>(((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and ceas$5 and provid$5</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/23 10:19</td></tr>
<tr><td>7</td><td>4</td><td>(("5285494") or ("5333183") or ("6405251") or ("6589078")).PN.</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/23 11:32</td></tr>
</table>

<table>
<tr><td>8</td><td>6</td><td>((((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and (ceas$5 or stop$5) and provid$5) and database and recover$5</td><td>USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB</td><td>2003/08/23 11:07</td></tr>
</table>

9	6	(((((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and (ceas$5 or stop$5) and provid$5) and database and recover$5) and monitor$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:07

10	0	(((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and ceas$5 and provid$5) and fault	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:08

<table>
<tr><td>11</td><td>0</td><td>(((((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and (threshold or predetermined)) and audit$5) and ceas$5 and provid$5) and (fault or error)</td><td>USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB</td><td>2003/08/23 11:08</td></tr>
<tr><td>12</td><td>4</td><td>(("5285494") or ("5333183") or ("6405251") or ("6598078")).PN.</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/23 11:48</td></tr>
<tr><td>13</td><td>4121</td><td>709/224</td><td>USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB</td><td>2003/08/23 11:49</td></tr>
</table>

14	3881	709/223	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:49
15	5775	709/203	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:49
16	742	455/424	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:49
17	1953	379/112	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/23 11:49
-	105	aggregat$3 near1 records	USPAT; US-PGPUB; EPO; JPO	2003/08/22 07:10
-	0	aggregat$3 near1 (related adj records)	USPAT; US-PGPUB; EPO; JPO	2002/05/07 11:30
-	17	(available near1 access near1 object) and related	USPAT; US-PGPUB; EPO; JPO	2002/05/07 11:30
-	76	(aggregat$3 near1 records) and related	USPAT; US-PGPUB; EPO; JPO	2002/05/07 11:31
-	56	((aggregat$3 near1 records) and related) and collect$3	USPAT; US-PGPUB; EPO; JPO	2002/05/07 11:31
-	21	(((aggregat$3 near1 records) and related) and collect$3) and identifier and structure and data	USPAT; US-PGPUB; EPO; JPO	2003/01/06 14:06
-	14	((((aggregat$3 near1 records) and related) and collect$3) and identifier and structure and data) and account$3	USPAT; US-PGPUB; EPO; JPO	2002/05/07 11:37
-	8	((((aggregat$3 near1 records) and related) and collect$3) and identifier and structure and data) and accounting	USPAT; US-PGPUB; EPO; JPO	2003/01/06 14:26
-	287	aggregation and distribution and receiving and data and entity and records	USPAT; US-PGPUB; EPO; JPO	2002/05/07 13:50
-	136	aggregation and distribution and receiving and data and entity and records and producing	USPAT; US-PGPUB; EPO; JPO	2002/05/07 13:51
-	53	aggregation and distribution and receiving and data and entity and records and producing and accounting	USPAT; US-PGPUB; EPO; JPO	2002/05/07 13:51

-	50	aggregation and distribution and receiving and data and entity and records and producing and accounting and structure	USPAT; US-PGPUB; EPO; JPO	2002/05/07 14:15
-	32	aggregation and distribution and receiving and data and entity and records and producing and accounting and structure and attributes	USPAT; US-PGPUB; EPO; JPO	2002/05/07 14:16
-	21	aggregation and distribution and receiving and data and entity and records and producing and accounting and structure and attributes and packets	USPAT; US-PGPUB; EPO; JPO	2002/05/07 14:16
-	8	network near1 activity near1 records	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:37
-	2	(("6032147") or ("5778350")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/01/06 14:07
-	14	((((aggregat$3 near1 records) and related) and collect$3) and identifier and structure and data) and accounting	USPAT; US-PGPUB; EPO; JPO	2003/03/29 15:34
-	2	(("6032147") or ("5778350")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/02/25 14:42
-	1	("5862344").PN.	USPAT; US-PGPUB; EPO; JPO	2003/02/25 14:42
-	16	((((aggregat$3 near1 records) and related) and collect$3) and identifier and structure and data) and accounting	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/04/02 13:20
-	8	("4135241" | "4667292" | "4858121" | "5229584" | "5606497" | "5713350" | "5732401" | "5799286").PN.	USPAT	2003/03/29 15:39
-	12	("5109486" | "5230048" | "5668955" | "5757798" | "5815556" | "5926104" | "6002948" | "6038551" | "6069941" | "6199195" | "6230203" | "6327049").PN.	USPAT	2003/03/29 15:42
-	684	internet same account$4 same record	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/04/02 12:56
-	52	(internet same account$4 same record) and aggregat$5 and filter$4 and (policy or rule) and collect$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/04/02 12:58
-	6	((((aggregat$3 near1 records) and related) and collect$3) and identifier same structure same data) and account$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/04/18 09:55

-	3	(("5557746") or ("6025447") or ("6418415")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/04/18 10:20
-	3	(("5557746") or ("6052447") or ("6418415")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/08/21 08:57
-	73	(("3463222") or ("4396058") or ("4449573") or ("4744410") or ("5020058") or ("5025457") or ("5109486") or ("5230048") or ("5465206") or ("5592620") or ("5630125") or ("5634009") or ("5668955") or ("5757798") or ("5761502") or ("5784443") or ("5793853") or ("5794221") or ("5799321") or ("5802502") or ("5815556") or ("5852812") or ("5920847") or ("5926104") or ("5949782") or ("5956391") or ("5958009") or ("5978780") or ("5991746") or ("5999604") or ("6002948") or ("6009154") or ("6014691") or ("6038551") or ("6047051") or ("6047268") or ("6058380") or ("6069941") or ("6070192") or ("6078907") or ("6078957") or ("6088344") or ("6088706") or ("6088789") or ("6091737") or ("6112236") or ("6118936") or ("6151601") or ("6192408") or ("6199195") or ("6230203") or ("6272126") or ("6275953") or ("6278995") or ("6282267") or ("6295532") or ("6321326") or ("6327049") or ("6359976") or ("6377567") or ("6381306") or ("6385301") or ("6418467") or ("5293379") or ("5430709") or ("5923659") or ("5931913") or ("6119160") or ("6157648") or ("6304892") or ("6308148") or ("6415312") or ("6452915")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/07/29 13:13

-	3	((("3463222") or ("4396058") or ("4449573") or ("4744410") or ("5020058") or ("5025457") or ("5109486") or ("5230048") or ("5465206") or ("5592620") or ("5630125") or ("5634009") or ("5668955") or ("5757798") or ("5761502") or ("5784443") or ("5793853") or ("5794221") or ("5799321") or ("5802502") or ("5815556") or ("5852812") or ("5920847") or ("5926104") or ("5949782") or ("5956391") or ("5958009") or ("5978780") or ("5991746") or ("5999604") or ("6002948") or ("6009154") or ("6014691") or ("6038551") or ("6047051") or ("6047268") or ("6058380") or ("6069941") or ("6070192") or ("6078907") or ("6078957") or ("6088344") or ("6088706") or ("6088789") or ("6091737") or ("6112236") or ("6118936") or ("6151601") or ("6192408") or ("6199195") or ("6230203") or ("6272126") or ("6275953") or ("6278995") or ("6282267") or ("6295532") or ("6321326") or ("6327049") or ("6359976") or ("6377567") or ("6381306") or ("6385301") or ("6418467") or ("5293379") or ("5430709") or ("5923659") or ("5931913") or ("6119160") or ("6157648") or ("6304892") or ("6308148") or ("6415312") or ("6452915")).PN.) and response and function$5 and instantiat$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/07/29 10:52
-	4	(("5557746") or ("6052447") or ("6418415") or ("6405251")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:05
-	4	(("5557746") or ("6052447") or ("6418415") or ("6405251")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:13
-	0	((("5557746") or ("6052447") or ("6418415") or ("6405251")).PN.) and collect$5 and filter$5 and stor$5 and output$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:21
-	5	bullard.inv. with william	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:21
-	0	(((("5557746") or ("6052447") or ("6418415") or ("6405251")).PN.) and collect$5 and filter$5 and stor$5 and output$5) and collect$5 and filter$5 and stor$5 and output$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:37
-	0	(((("5557746") or ("6052447") or ("6418415") or ("6405251")).PN.) and collect$5 and filter$5 and stor$5 and output$5) and collect$5 and filter$5 and stor$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:21

-	5	bullard.inv. with william	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:21
-	0	(bullard.inv. with william) and collect$5 and filter$5 and stor$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:22
-	4	(bullard.inv. with william) and collect$5 and aggregat$5 and stor$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:25
-	4	(bullard.inv. with william) and collect$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:25
-	0	((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and filter$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:25
-	2	((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and allow$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:26
-	2	(((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and allow$5) and report$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:26
-	0	((((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and allow$5) and report$5) and submit$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:26
-	2	((((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and allow$5) and report$5) and quer$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:27
-	2	(((((bullard.inv. with william) and collect$5 and aggregat$5 and stor$5) and allow$5) and report$5) and quer$5) and complet$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:27
-	6	("4538030" | "4979207" | "5027388" | "5159698" | "5325290" | "5333184").PN.	USPAT	2003/08/21 09:35

-	299	network with activity with record	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:39
-	37	(network with activity with record) and collect$5 and filter$5 and stor$5 and output$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:40
-	15	((network with activity with record) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 09:40
-	298	network with usage with record	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 09:40
-	61	(network with usage with record) and collect$5 and filter$5 and stor$5 and output$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 10:24
-	29	((network with usage with record) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5	USPAT; US-PGPUB; EPO; JPO	2003/08/21 10:27
-	1	("6317884").PN.	USPAT; US-PGPUB; EPO; JPO	2003/08/22 07:29

-	110	("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.	USPAT	2003/08/21 10:23

-	24	(("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5	USPAT;
			US-PGPUB;
			EPO; JPO
				2003/08/21
				10:25

<table>
<tr><td>-</td><td>15</td><td>((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/21 10:31</td></tr>
</table>

<table>
<tr><td>-</td><td>5</td><td>(((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and (network adj usage)</td><td>USPAT; US-PGPUB; EPO; JPO</td><td>2003/08/21 11:03</td></tr>
</table>

<table>
<tr><td>-</td><td>12</td><td>(((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and internet</td><td>USPAT;
US-PGPUB;
EPO; JPO</td><td>2003/08/21
11:04</td></tr>
</table>

-	15	(((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network	USPAT; US-PGPUB; EPO; JPO	2003/08/21 11:18

-	14	((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)	USPAT; US-PGPUB; EPO; JPO	2003/08/22 07:21
-	5150	load adj balancing	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 13:55
-	970	(load adj balancing) same performance	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 13:56

-	304	((load adj balancing) same performance) and (application adj program)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 13:57
-	130	(((load adj balancing) same performance) and (application adj program)) and policy	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 13:57
-	8	((((load adj balancing) same performance) and (application adj program)) and policy) and (server adj farm)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/21 13:57
-	3175	aggregat$3 near1 (record or data)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:10
-	333	(aggregat$3 near1 (record or data)) and filter$5 and allow$5 and complet$5 and submit$5 and (output$5 or produc$5)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:11
-	15	((aggregat$3 near1 (record or data)) and filter$5 and allow$5 and complet$5 and submit$5 and (output$5 or produc$5)) and collect$5 and (real-time or realtime) and stor$5 and error and threshold and audit$5 and bill$6 and account$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:13
-	15	(((aggregat$3 near1 (record or data)) and filter$5 and allow$5 and complet$5 and submit$5 and (output$5 or produc$5)) and collect$5 and (real-time or realtime) and stor$5 and error and threshold and audit$5 and bill$6 and account$5) and report$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:15
-	12	((((aggregat$3 near1 (record or data)) and filter$5 and allow$5 and complet$5 and submit$5 and (output$5 or produc$5)) and collect$5 and (real-time or realtime) and stor$5 and error and threshold and audit$5 and bill$6 and account$5) and report$5) and usage	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:15

-	14	((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)	USPAT; US-PGPUB; EPO; JPO	2003/08/23 09:54
-	101	herbert.inv. with james	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:22
-	2	(herbert.inv. with james) and record and usage and collect$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:25

<table>
<tr><td>-</td><td>14</td><td>(((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and report$5</td><td>USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB</td><td>2003/08/22 07:27</td></tr>
</table>

-	14	(((((("4464543" | "4529842" | "4608693" | "4935956" | "4972453" | "4987538" | "4999833" | "5014267" | "5285494" | "5287270" | "5325290" | "5333183" | "5341377" | "5347305" | "5357564" | "5461611" | "5515425" | "5559527" | "5629938" | "5650994" | "5652787" | "5666481" | "5684965" | "5694548" | "5696906" | "5742762" | "5745883" | "5761650" | "5774532" | "5790634" | "5790642" | "5799072" | "5799073" | "5802160" | "5809120" | "5822744" | "5828737" | "5835081" | "5842183" | "5852659" | "5862203" | "5862329" | "5864823" | "5867494" | "5867495" | "5867653" | "5878130" | "5881131" | "5892754" | "5893077" | "5896445" | "5898688" | "5905715" | "5907607" | "5913036" | "5915008" | "5917817" | "5917898" | "5918213" | "5930333" | "5933425" | "5933489" | "5940471" | "5943657" | "5950174" | "5953389" | "5956720" | "5960416" | "5966698" | "5970065" | "5970403" | "5978855" | "5982362" | "5987107" | "5987430" | "5991378" | "5991381" | "5991806" | "5995606" | "5995946" | "5999525" | "5999604" | "6005926" | "6006205" | "6011838" | "6026379" | "6031904" | "6034950" | "6044355" | "6044362" | "6052447" | "6052671" | "6055240" | "6070150" | "6081518" | "6101481" | "6119109" | "6134223" | "6147975" | "6157648" | "6216169" | "6195697" | "6345239" | "6426948" | "6427132" | "6434628" | "6438594" | "6442547" | "6442748" | "6449855").PN.) and collect$5 and filter$5 and stor$5 and output$5) and aggregat$5) and network) and (real-time or realtime)) and output$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/22 07:27
-	2	(("6317884") or ("6418467")).PN.	USPAT; US-PGPUB; EPO; JPO	2003/08/22 08:50

102(e)

US-PAT-NO: 5333183

DOCUMENT-IDENTIFIER: US 5333183 A

TITLE: Universal MDR data record collection and reporting system

DATE-ISSUED: July 26, 1994

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Herbert; James H.	Pittsford	NY	N/A	N/A

APPL-NO: 07/ 851553

DATE FILED: March 13, 1992

US-CL-CURRENT: 379/112.07, 379/112.01, 379/126

ABSTRACT:

An MDR data record collection and reporting system collects MDRs of various formats from PBX and telephone company Centrex switches of different types, for multiple customers. A set of switches is monitored by a monitor. The monitor includes one message processor connected to each switch, and an administrator processor, which is connected to all message processors for that set of switches. Each message processor captures MDRs from the switch to which it is connected. After the MDR has been captured, the message processor reformats the MDR from a format used on that type of switch into a selected format. The message processor also collects and sends statistics and alarms to the administrator processor. The administrator processor prepares statistical reports and alarm reports. Multiple monitors can be included within a telephone

system. A collector is provided which can receive reformatted MDRs from one or more monitors, and can store reformatted MDRs for customers with sets of switches at multiple locations. A manager is provided which can receive reformatted MDRs from one or more monitors and one or more collectors. The manager and collector use the reformatted MDRs to prepare various reports.

38 Claims, 20 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 17

Abstract Text - ABTX (1):

An MDR data record collection and reporting system collects MDRs of various formats from PBX and telephone company Centrex switches of different types, for multiple customers. A set of switches is monitored by a monitor. The monitor includes one message processor connected to each switch, and an administrator processor, which is connected to all message processors for that set of switches. Each message processor captures MDRs from the switch to which it is connected. After the MDR has been captured, the message processor reformats the MDR from a format used on that type of switch into a selected format. The message processor also collects and sends statistics and alarms to the administrator processor. The administrator processor prepares statistical reports and alarm reports. Multiple monitors can be included within a telephone system. A collector is provided which can receive reformatted MDRs from one or more monitors, and can store reformatted MDRs for customers with sets of switches at multiple locations. A manager is provided which can receive reformatted MDRs from one or more monitors and one or more collectors. The manager

and collector use the reformatted MDRs to prepare various reports.

US Patent No. - PN (1):

5333183

TITLE - TI (1):

Universal MDR data record collection and reporting system

Brief Summary Text - BSTX (3):

The invention relates to a data record collection and reporting system for use with PBX and telephone company switches, and, more particularly, to an MDR collection and reporting system which collects MDRs of various formats from switches of different types for multiple telephone company customers.

Brief Summary Text - BSTX (5):

Telephone company customers such as corporations, government agencies, and institutions often have multiple locations served by private telecommunications networks. Multiple telephone company customers are in a network served by a telephone company. Several telephone company central office switches are used to provide telephone service for a geographic area.

Brief Summary Text - BSTX (6):

Many customers elect to provide telephone service within a location themselves, using a Private Branch Exchange (PBX), one form of an intelligent network node.

Brief Summary Text - BSTX (7):

Others elect to have telephone service provided by their telephone company through a service called Centrex, another form of an intelligent network node. Centrex provides business services to many customers from each central office switch. The public network telecommunications provider (a telephone company) may provide Centrex service to multiple organizations from each central office switch.

Brief Summary Text - BSTX (10):

Demand for telecommunication services is constantly increasing, making it increasingly important for organizations to control their costs. The customer's need to control the costs associated with provider telecommunications has introduced the requirement of collecting detailed information on the usage of the network.

Brief Summary Text - BSTX (12):

Many different types of intelligent nodes and telephone switches are available on the market to serve different needs of customers. Unfortunately, each type of node and switch provides the information about each call in a different format. There exists a profusion of different MDR formats, all used within a telephone network.

Brief Summary Text - BSTX (13):

The public network telecommunications provider has multiple switches. Certain large customers may use multiple central office switches. Because the public network telecommunications provider provides Centrex to multiple organizations from multiple central office switches, it needs to be able to collect usage information for multiple customers on each of the multiple switches. Further, it needs to collect each MDR separately for each customer.

Brief Summary Text - BSTX (14):

There further exists a need for the public network telecommunications provider to deliver usage information to each customer reliably and rapidly.

Brief Summary Text - BSTX (15):

The organization's need to control the costs associated with provider telecommunications has introduced the requirement of collecting detailed information on the usage of the private telecommunications network. Thus, there exists a need for the customer to be able to further process the usage information to provide administrative, planning and maintenance functions regarding the customer's network.

Brief Summary Text - BSTX (18):

Conventional systems lack these features. For example, one conventional system is shown in U.S. Pat. No. 4,656,656, issued Apr. 7, 1987, to Mundy, Jr. et al. In Mundy, a hospital call accounting system is described that collects MDRs, which Mundy calls CDRs. The MDRs, however, are collected from a single PBX system. Thus, Mundy has multiple disadvantages in that it works only with a single customer, is limited to PBX, and is limited to a single switch.

Brief Summary Text - BSTX (19):

Another conventional system is shown in U.S. Pat. No. 4,464,543, issued Aug. 7, 1984, to Kline et al. Kline discloses a call trace facility built into a network control center for a private branch exchange. Multiple MDRs are collected from multiple switches. Unfortunately, Kline is limited to use with PBX switches, and cannot be used with Centrex. Furthermore, Kline uses the MDRs primarily to trace calls. Thus, Kline does not provide other information in the MDR in a format useful for network administrative, planning and maintenance functions. A further drawback of Kline is that Kline does not deal with multiple customers.

Brief Summary Text - BSTX (20):

Another example of a conventional system is shown in U.S. Pat. No. 4,525,601, issued Jun. 25, 1985, to Barnich et al. Barnich discloses a system for efficient costing of telephone calls, by collecting MDRs, which Barnich calls SMDRs, from a PBX. Unfortunately, Barnich does not deal with multiple switches, Centrex, or multiple customers.

Brief Summary Text - BSTX (21):

Yet another conventional system is shown in U.S. Pat. No. 5,048,079, issued Sep. 10, 1991, to Harrington, et al. Harrington discloses enhancing the accuracy of call duration information in an MDR, which Harrington calls SMDRs, by inserting an apparatus into a trunk of a PBX. The apparatus collects MDRs from the PBX and outputs them with enhanced call duration accuracy. However, Harrington does not deal with multiple switches, Centrex, or multiple customers.

Brief Summary Text - BSTX (22):

Finally, U.S. Pat. No. 4,788,718, issued Nov. 29, 1988, to McNabb, et al. that essentially generates MDRs by monitoring certain messages, called SS7 messages. McNabb can function with multiple customers, however, McNabb has the distinct disadvantage that it does not use MDRs generated by PBX or Centrex.

Brief Summary Text - BSTX (24):

Telesciences produces a passive product that emulates a tape drive and collects AMA data and transmits it to some telco provided billing or collection system. They have produced an adjunct to that system designed to allow the parcing of that data flow to create an MDR like record from the AMA data stream. This product does not produce a Bellcore standard MDR record. It does not contain call records for anything other than billable calls. It does no real time accuracy or integrity checking, and it does not conform to any open architecture standards.

Brief Summary Text - BSTX (25):

AT&T produces three products that provide SMDR information to customers; the ACP, BCP and an MDR Translator that provide SMDR to premise directly from the central office switch to a single customer only. The ACP and BCP are mini computer architectured systems that are not built to reside within a central office, are intended to support more than one central office, do not do any edit checking of call records, do not produce call records that conform to any standard, are not integrated into a call record delivery system for the end user customer, and do not fit into a Collector system architecture.

Brief Summary Text - BSTX (28):

An object of the invention is to provide an MDR data record collection and reporting system which can be used with both PBX and Centrex switches of all types.

Brief Summary Text - BSTX (31):

The universal MDR data record collection and reporting system receives an MDR in one of a plurality of MDR formats from one of a plurality of types of switches on a telephone network. It reformats the MDR to one of a plurality of selected formats. It also stores the reformatted MDR in an MDR file. Further, it also

transmits the reformatted MDRs from the message processor using a communications protocol.

Drawing Description Text - DRTX (7):

FIG. 4 is a block diagram of a monitor, showing multiple message processors, connected to switches of different types, which could be produced by different vendors, and an administrator processor;

Drawing Description Text - DRTX (8):

FIG. 5 is a block diagram of communication between the monitor and a collector;

Drawing Description Text - DRTX (9):

FIG. 6 is a block diagram of communication between the monitor and a manager;

Drawing Description Text - DRTX (10):

FIG. 7 is a block diagram of multiple message processors communicating with a manager and multiple message processors communicating with a collector;

Drawing Description Text - DRTX (11):

FIG. 8 is a block diagram of multiple message processors and a simplified MDR capture device communicating with a manager and a collector;

Drawing Description Text - DRTX (12):

FIG. 9 is a block diagram of a telecommunications system of FIG. 1 using a universal MDR data record collection and reporting system, including a monitor, a manager and a collector;

Drawing Description Text - DRTX (17):

FIG. 14 is a block diagram of the message processor collection of an MDR;

Drawing Description Text - DRTX (23):

FIG. 20 illustrates collecting and delivering MDRs.

Detailed Description Text - DETX (2):

A universal MDR data record collection and reporting system for use with Centrex switches or capture devices used with PBXs on a telephone network includes a message processor with a receiver, receiving an MDR in one of a plurality of M DR formats from one of a plurality of types of switches on a telephone network. The message processor also has a reformatter, reformatting the MDR to one of a plurality of selected formats. Additionally, the message processor has a means for storing the reformatted MDR in an MDR file. Further, the message processor also has a transmitter, transmitting the reformatted MDRs from the message processor on a communications link using a communications protocol.

Detailed Description Text - DETX (3):

FIG. 1 is an example of a network with multiple customers served by a telephone company. In the example, a plurality of telephone company central office switches 1 are used to provide telephone service for a geographic area. Many customers elect to provide telephone service within a location using a PBX 2. Others elect to have telephone service provided by their telephone company through a service called Centrex which provides business services to many customers from each central office switch 1. As illustrated, a medium to large sized customer will typically have several locations, some served by Centrex and others served by PBX. See, for example, Customers A and B. The Central Office may be connected to the PBX by a CO-PBX trunk 600. Customers often have private telecommunications networks comprised of the PBX or Centrex intelligent nodes, a tie trunk 3 that interconnects the nodes and a plurality of telephone sets 4, which are also called stations.

Detailed Description Text - DETX (4):

The universal MDR data record collection and reporting system includes a message processor, which is connected to one switch. A message processor is capable of collecting MDRs from switches using the following types of switch interfaces: 5ESS, available from AT&T; DMS-100, available from Northern Telecom; 1/1 AESS, available from AT&T; and compatible switch interfaces. The message processor can handle an MDR format according to the following specifications, incorporated herein by reference: 5ESS Switch ISDN Interface Applications Processor Interface Specification, Issue 4.00, Oct. 1990, AT&T 235-900-303; Output Specification Basic Communications Package Version 3, Issue 2, May 1990, AT&T 533-615-203; Northern Telecom AMAB150, release 09.01, Practice 297-1001-510; Northern Telecom Digital Switching Systems DMS-100 Integrated Business Network (IBN) Station Message Detail Recording Reference Manual, release 06.01 standard, issued Sep. 25, 1987, Practice 297-2001-119; Northern Telecom Digital Switching System Network Operations Systems Business Network Management SMDR Interface Specification, vintage NSR28 02, issued Nov. 10, 1989, compatible with NIS Q.202-1, Practice 450-1021-181; Bell Communications Research Message Detail Recording (MDR), FSD 02-02-1110, Technical Reference TR-TSY-000610, Issue 1, Jul. 1990; Bell Communications Research, Record Format for Transmitting Message Detail Recording Data from a Central Office Switching System, FSD 02-02-1120, Technical Reference TR-TSY-000620, Issue 1, Jul. 1990; Bell Communications

Research, Description of the Interface Between a 1/1 AESS Switching System and a Customer Premises Message Detail Recording System, Technical Reference TR-533-23112-84-01, Issue 1, May 1984; and Output Specifications, Basic Communications Package, Version 3, Issue 2, May 1990, AT&T 533-615-203. FIG. 2 illustrates a capture device 17 connected to a PBX 2. FIG. 3 illustrates a message processor 5 connected to a Central Office Switch 1.

Detailed Description Text - DETX (5):

As illustrated in FIG. 14, the message processor collects an MDR in the following manner. A telephone call is transmitted on the switch. When the call is completed, the switch prepares the MDR. The switch then transmits the MDR over a link 40 to a receiver 42, using a protocol appropriate to the switch, for example, X.25 or RS-232. The receiver 42 receives the MDR from the link 40. The MDR is reformatted by a reformatter 44 into an intermediate format. The intermediate format can be selected by a customer. The selected format is stored in configuration storage 54, and accessed by the reformatter 44. The message processor includes a disk 46, so that the reformatted MDR is stored on the disk and can later be retrieved. The message processor includes a separate disk file for each customer for which it receives MDRs. Thus, MDRs for each customer are segregated from other customers' MDRs on the same disk.

Detailed Description Text - DETX (6):

In addition to reformatting the MDR, the message processor includes an error detector 48 which detect errors in MDRs. Furthermore, the message processor counts and stores statistics in a statistics collector 50 for the switch. The message processor also detects errors internal to itself and stores them as alarms in an alarm collector 52.

Detailed Description Text - DETX (9):

Referring now to FIG. 4, multiple switches, which can be of different types, 1a, 1b, 1c, manufactured by different vendors, are connected to message processors 5. Each message processor 5 is connected to a single administrator processor 8. The message processors 5 and their administrator processor 8 are collectively referred to as a monitor 15.

Detailed Description Text - DETX (10):

The administrator processor 8 is used by the telephone company. The administrator processor is the recipient of alarms from each of the message processors and reports them to telephone company personnel responsible for resolving the problem. It is also the system into which the configuration is entered by telephone company personnel when a new MDR customer is added, or when the configuration changes. It later loads the configuration into each message processor that handles the customer with changed configurations. The administrator processor has an Administrative Processor Interface (APIF) 81 which forms communications between the administration processor 8 and manager 6, collector 7 and other elements, as needed. This is accomplished by downloading communications information from the administrator processor to the message processors.

Detailed Description Text - DETX (11):

In addition to being accessed by the administrator processor, the stored reformatted MDRs are accessible by a system at a central site using a specified protocol. This is preferably implemented by the message processor communicating with the central site system. The message processor transmits reformatted MDRs to the central site. The central site could be a customer site or a telephone company office.

Detailed Description Text - DETX (12):

Referring now to FIG. 5, one possible central site is a collector. The collector 7 receives MDRs over a transmission link 16 from several message processors 5 so that MDRs for customers with multiple locations are stored in one location. The collector obtains the MDR from the monitors in real-time if dedicated transmission links are available and uses a dial-up polling arrangement otherwise. The collector stores the MDR in a separate disk file for each customer and switch. The MDRs may be transmitted to customer's central site in real-time if a dedicated transmission link is available, otherwise they are transmitted when the collector is polled for the MDR. The collector is preferably located in a telephone company building.

Detailed Description Text - DETX (13):

Referring now to FIG. 6, another possible configuration is a manager located at a customer site. The manager 6 receives MDRs over the transmission link 16 from the monitor 15. Alternatively, as shown in FIG. 7, the manager also receives MDRs over the transmission link from the collector if the customer has multiple locations served by central offices using the same collector. The manager obtains the MDR in real-time if there is a dedicated transmission link to the message processor or collector. If a dedicated transmission link is not used, the manager establishes a dial connection to the appropriate message processor or collector to collect the MDR on a polling basis. The manager then uses the MDR to cost calls for a departmental telephone bill or for other management purposes. The manager is preferably located at a customer's location.

Detailed Description Text - DETX (14):

FIG. 8 shows the collection device 17 connected to a switch, for example, a PBX 2. The collection device has rudimentary ability to capture and store MDRs. The collector 7 or manager 6 can obtain stored MDRs from the collection device by a transmission link 16, using real-time if the transmission link 16 is dedicated, and using a polling arrangement otherwise.

Detailed Description Text - DETX (15):

FIG. 9 illustrates one example of a universal MDR collection/reporting system on a telephone network corresponding to FIG. 1. In this example, the system includes the monitor having several message processors 5 and one administrator processor 8. In this example, the transmission link from the message processor 5 to the administrator processor 8 is a dial-up telephone line as opposed to a dedicated link. Therefore, in FIG. 9, the administrator processor 8 is not connected. The system of this example also includes one collector 7 and one manager 6. In this example, the manager 6 only communicates with the collector 7. The collector 7 communicates with the monitor via the message processors 5. One PBX includes a collection device 17, with which the collector 7 communicates. Customer B has two PBXs connected by a tie trunk 3. Customer A is connected by two tie trunks 3 to two central offices.

Detailed Description Text - DETX (18):

The message processor CPU card 60 is based on a 32 bit external 68EC030 processor operating at 25 MHz. This processor provides the message processor with significant data handling capability. One or 4M bytes of dynamic RAM is available for program execution and temporary storage. A plurality of VLSI chips provide the full VME bus capability including arbitration, interrupt generation and handling, and data block transfers. Preferably, two full duplex, asynchronous, RS-232 ports are accessed through RJ-11 connectors on a front panel of the CPU board.

Detailed Description Text - DETX (19):

The side card 62 is connected to the bottom of the CPU card by a modular connector. The side card my be directly below the CPU card when they are mounted in the message processor shelf. This card supports the central office switch interface and my therefore have several variations. Referring now to FIG. 16, it should include a NCR 700 series or equivalent SCSI disk controller 60 capable of working in a simplex or mirrored disk environment, a real time clock providing calendar and clock functionality, up to 32 kbyte of battery backed CMOS static RAM 62 and a software controlled watchdog timer 64. The side card has a MC68302 processor to handle communication protocols including ISDN/BRI OB+D, X.25 LAP D and a single port on the front panel of the board providing ISDN for the 5ESS switch and X.25 LAP A communications for the 1AESS switch. Also included are a state machine 70, command register 72, alarm timer 74, and bus gasket 76.

Detailed Description Text - DETX (20):

Storage is provided on a disk 64. A standard 3.5" SCSI hard disk such as known to those of ordinary skill in the art is provided. The disk is mounted on a commercially available size, two width (1.6"), double eurocard module. The SCSI interface uses the P2 connector to communicate between the CPU board and a disk subsystem. Disk drives of 60, 100, and 210 MB are used, depending on required storage capacity. Up to three disk subsystems (for a total of four) can be added to the system to increase storage capacity. Other standard size drives could be substituted. Each hard disk subsystem is placed in the chassis in a predesignated slot (simplex versus mirrored) and contains LED indicators showing disk activity and a fault light that indicates that the CPU has determined that the drive is no longer reliable. Disks can be removed/inserted into a running system.

Detailed Description Text - DETX (29):

It is further preferable that the operating system has a kernel providing multi-tasking, interprocess communication, memory management, and input-output, and a set of utilities for housekeeping, management and customization of software. OS-9 consists of the kernel (which provides multi-tasking, inter-process communication, memory management and input-output) and a set of utilities (for housekeeping, management and customization of the software environment). Its system interfaces are very similar to those of the UNIX.RTM. operating system.

Detailed Description Text - DETX (32):

This disclosure herein describes one embodiment of the invention where the message processor collects an MDR from an AT&T 5ESS.RTM. switch and delivers it in one of the selected formats.

Detailed Description Text - DETX (33):

We will now describe the steps involved in collecting an MDR from the switch 1 and delivering it to the customer's manager 6. The message processor 5 is first involved when a call on the switch 1 has been completed and the MDR prepared by the switch. The 5ESS switch transmits the MDR data on a ISDN link using D channel packetized data. As the receiver 42, the Switch Interface process A5 retrieves the record from the X.25 packet assembler/disassembler. As the reformatter 44, it then examines the MDR to determine which customer's MDR was received and converts it into the desired output format, for example, Ameritech Call Detail Recording System or Bellcore AMA Format, selected for that customer. The reformatted MDR is then written into one of a series of disk files on disk 46. Preferably, the disk file is suitable for later transmission using a file transfer communications protocol, such as Kermit. The statistics for the switch and that customer are updated. The Switch Interface process is described in more detail below.

Detailed Description Text - DETX (35):

The message processor must collect statistics which are used, for example, to ensure that an adequate number of communications links are available to handle switch to message processor MDR traffic and to decide when more links should be added. Therefore, in this embodiment, a Statistics Process A6 acts as a statistics collector and collects statistics for use in producing management reports for the telephone company. The Statistics process runs on a scheduled basis and periodically, for example, each hour transmits a request to send statistics for the previous time period to the Switch Interface process and Kermit File Builder processes (as well as to all single direct customer premises interfaces A8 and multi-direct customer premises interfaces A9). Periodically, the administrator processor should log on to each message processor and collect statistics. Statistics my be collected by translation intoASCIi and transmitted by the message processor Data Unloader process.

Detailed Description Text - DETX (37):

When the message processor leaves the factory, the OS-9 operating system is in ROM and the Installation Process A1 is on the hard disk. When the message processor is turned on for the first time, the Installation Process A2 initializes modem software and then starts a known OS-9 tsmon process which enables an administrator processor communications port for login purposes. The administrator processor logs onto the message processor and uses the OS-9 command interpreter to run utility programs to download the software and update database records as needed. After an initial installation of software, the Initialization Process A2 is started. Subsequently, it will be started automatically when the message processor is restarted.

Detailed Description Text - DETX (38):

The Switch Interface process is described with reference to the flow chart in FIG. 11. At step 100, the Switch Interface process starts when it is initiated by the Initialization Process A2. At step 101, the 5ESS version of this software first initializes an ISDN packet driver by calling a known ss.sub.-- opt() and a known isdn.sub.-- cfg() functions. The ISDN packet driver is the interface to the MDRs from the 5ESS which are carried as ISDN D channel packet data. (The DMS-100.TM. version of this process is described later). Next, at step 102, a named pipe is created using the known open() system call to receive commands. At step 103, the customer database is read and a customer information list entry built for each customer. The statistic counters are then initialized at step 104. The final initialization step, 105, is to initialize the MDR file interface by using known operating system calls to establish the Switch Interface process as a writer for the files, and to establish the linkages between customers and their MDR files.

Detailed Description Text - DETX (40):

If an MDR was received, at step 115, MDRs are read from the ISDN packet driver by the MDR Reader which increments the count of MDRs received from the switch and checks to see if the MDR contains a valid customer ID. MDRe which are in error as determined in step 107a are stored so that they my be reprocessed after the error is fixed. Therefore, there may be an Errored MDR file to store MDRs which are in error. Furthermore, there may also be a file, for example, an errc.dat file, for storing a small number of MI)Rs which are in error. The file my be examined by an operator, so that the operator can determine the cause of the error, and then can fix the cause of the error. Therefore, if an error was returned by the MDR Reader, as determined in step 107a, a check is done to see if the Errored MDR files are full at step 116. If not, the Errored MDR Handler is called at step 118 to write a copy of the MDR (as received from the switch) to the Errored MI)R File and another copy to the errc.dat file if this file contains less than 25 MDRs. If an error was not returned from the MDR Reader, the MDR writer is called at step 108.

Detailed Description Text - DETX (41):

The MDR Writer's logic is described with reference to FIG. 12. At step 200, the MDR writer starts when it is called from the Switch Interface process. At step 201, it extracts the Customer ID and MI)R type from the MDR received from the switch. At step 202, it uses these to look up the MDR conversion routine for that customer and switch and, at step 203, checks to see if there was an error. If not, at step 204 the conversion routine just looked up is called to convert the MDR into the output format selected for the customer owning the MDR. At step 216, if the conversion was successful, (no errors found in step 205 and no overflow detected at step 206) the MDR is written to the customer MDR file and, at step 209, the MDR writer returns with a success status. If writing that MDR would have caused an overflow, at step 207, the Overflow Handler is called to write the MDR to the overflow file and, at step 208, the MDR Writer returns with an overflow status code. At step 210, if step 205 determined that there were errors finding the conversion routine or converting the MDR to the customer's format, the switch error count for the period is incremented. At step 211, if the incremented error count exceeds the threshold, the Alarm Handler is called. The Alarm Handler sends the error message to the Alarm Interface process. If the errored MDR queue is not full, as determined in step 212, at step 215, the Errored MDR Handler is called to write the MDR to the errored MDR file and, at step 213, the MDR writer returns with an Errored status.

Detailed Description Text - DETX (43):

Referring now to FIG. 11, if step 107 determined that the Switch Interface process received an IPC message at step 106, at step 117 it tests the message code in the IPC message. If the message was a Database change message, the Switch Interface process branches to repeat a portion of its initialization starting with reading in the modified database at step 103. If the message was a Send Statistics request, the transaction data information collected during the statistics collection period is sent to the Statistics Process in an IPC message at step 119 and then an end of statistics message is sent at step 122 to indicate the end of statistics data. The Switch Interface process then loops to read the next MDR or IPC message at step 106. If the message was a Stop message, the Switch Interface process exits at step 120.

Detailed Description Text - DETX (44):

In the case of the DMS-100, the MDR data is transmitted from the switch on either an asynchronous RS-232 interface that can be used to print the MDR or on a Pertec 9-track tape drive interface that can also be used to connect to an automatic message accounting data accumulator or a tape drive. Therefore, MDRs can be collected by simulation of these devices. In the DMS-100 version of Switch Interface process, the call to initialize the ISDN packet driver at step 101 is replaced with a call to initialize a communication port. The MDR Reader then reads the MDR from the communications port instead of from the ISDN

Packet driver.

Detailed Description Text - DETX (45):

Tables 7(a)-(e) describe MDR formats some of which are used by the message processor. Tables 7(a)-(c) describe reformatted MDRs, while Tables 7(d) and (e) describe MDR formats from the switch. The Bellcore AMA Format (BAF) was primarily intended for storing call billing information for use in billing telephone subscribers. The BAF has a modular design with separate modules for various purposes as shown in FIG. 7(a). Several of the existing BAF modules and a few additional ones were selected for use in MDR as shown in Table 7(b). MDR formats are determined by the type of switch to which the message processor is connected. Two popular kinds of switches are the AT&T 5ESS and the Northern Telecom DMS-100 whose MDR formats are shown in Table 7(d) and Table 7(e) respectively. The conversion routines called at step 204 in the MDR writer convert the MDR from the MDR format into the selected reformatted MDR.

Detailed Description Text - DETX (47):

The logic of the Kermit File Builder process is described in FIG. 13. The process first initializes by reading through a Customer Configuration table extracting a record for each Customer using Kermit, at step 301. If this is not the last customer, as determined in step 302, it then finds the disk with the largest amount of free disk storage at step 310 and builds a list of files for the customer at step 311. It then opens the MDR File for the customer at step 312. (This is the file into which the Switch Interface will simultaneously write MDR). It then opens the next Kermit file for the customer at step 313. (This is the file into which the Kermit File Builder process will write MDR). If all customers have been processed, as determined in step 302, the Kermit File Builder will then process IPC messages and MDR.

Detailed Description Text - DETX (49):

If an IPC message was received, the predefined message code in the IPC is tested at steps 314, 316. If the code was a predefined code KFB.sub.-- Xfer.sub.-- ID, the Kermit transfer of files to the customer's manager has finished. The Kermit File Builder then sends statistics on the transfer to the Statistics Process at step 315. If the code was a predefined code Send.sub.-- Stats.sub.-- ID, the Kermit File Builder sends storage statistics on disk storage for each customer as well as the total to the Statistics Process at step 317. If the IPC Code is Database-Change-ID, control returns to step 303.

Detailed Description Text - DETX (51):

If the files are to be sent to the collector 7 instead of directly to the manager, the process is similar to the manager calling the message processor.

Detailed Description Text - DETX (52):

Several processes are used to provide remote, centralized operations, administration and maintenance for the message processors operated by a telephone company. This is especially significant to telephone companies, as it enables them to run their network with a much smaller nun%her of personnel. The major areas that can be handled remotely are surveillance (i.e. the reporting of problems with the message processor or its interfaces), administration of message processor databases, updating the message processor software, and the analysis of data reporting the utilization of various message processor resources. These are handled both through message processor application software processes (e.g., the Alarm Interface process and the Statistics Process) and by utilities driven by the administrator processor over the communications line (e.g., the process and the message processor Data Entry Process).

Generally, these remote activities are done by transferring a file (with a known filename) to the message processor or administrator processor where it is processed by the software.

Detailed Description Text - DETX (53):

The Alarm Interface process A14 is responsible for receiving alarm information from the IPC message queue and writing it to an Alarm File in the format shown in Table 2. The Alarm Interface process then uses a communications port to dial an administrator processor's Alarm Port and transfers the Alarm File to the administrator processor where it is entered into a database and telephone company personnel, for example, a System Administrator, are notified. Alarms should be always sent though the administrator processor's Alarm Port even if there is a dedicated connection to the administrator processor for transferring table data and statistics. The administrator processor's AlarmPort 701 has an Alarm Requestor 703 for requesting and receiving alarms from the message processor and an alarm transmitter 705 for transmitting the alarms to database and telephone company personnel.

Detailed Description Text - DETX (54):

The Statistics Process A6 is responsible for requesting and collecting statistics from the processes that communicate with the switch and the customers. The Statistics Process is started by the Initialization Process and then runs on a scheduled basis. When a collection period ends, the Statistics Process sends a message to the other processes to signal them to send their current statistics and begin keeping statistics for the next collection period. The statistics are written to one of three statistics files as they are received from the IPC message queue.

Detailed Description Text - DETX (55):

A message processor Statistics File contains statistics on the overall message processor and the switch interface. Most of this information is collected by the Switch Interface process as it processes MDRs received from the switch and is counted at steps 110, 112, 114 into cells of a data structure that is sent to the Statistics Process when requested at the end of each statistics collection period. Each record in the file as shown in Table 3(b) below is copy of that structure.

Detailed Description Text - DETX (56):

A Customer Transfer Statistics File contains statistical information on MDRs transferred to a customer during a polling session or during a statistics collection period. The format of the information in the Customer Transfer Statistics file is shown in Table 3(a) below. This information is collected by the Kermit File Builder and sent to the Statistics Process upon request. Certain aperiodic processes (i.e. the Scheduled Customer Premises Interface and Polled Customer Premises Interface processes) will send statistics without being prompted when they have completed a transmission session. The Statistics Process consolidates information from these sources and prepares them for transmission to the administrator processor. The administrator processor has a statistics requestor 707 for requesting and receiving statistics from the message processor.

Detailed Description Text - DETX (57):

When the administrator processor 8, as shown in FIGS. 18 and 19, needs to update the message processor's tables, a communicator, such as an Administrator Processor Interface (APIF) 81 (the administrator processor process responsible for managing message processor communications) uses an administrator processor communications port 83 to dial the message processor communications port and log on using an administrative userid. When OS-9 accepts the logon, it runs the shell command interpreter. The APIF then uses the shell to execute a sequence of shell commands to do the update. The message processor Data Loader (not shown) is invoked to add information to message processor tables. The message processor Data Loader process reads ASCII records from its standard input (which are sent by the APIF) and loads them into the message processor table or data file as specified by the APIF. When it reaches the end of the file, the message processor Data Loader will print a success message on its standard output (which will be read by the APIF) and exit. The APIF then invokes the Database Update Notifier process which sends a database.sub.-- change IPC message to each process using the updated database. This would be received by the Switch Interface process and handled by repeating the portion of the Switch Interface process' initialization beginning at step 103 as described above.

Detailed Description Text - DETX (58):

The administrator processor 8 is a centrally located system used to create and maintain system configuration and user profile information for the central-office located message processors. The administrator processor's software may advantageously use the UNIX operating system which runs on various commercially available workstations. The preferred hardware embodiment is the Sun Microsystems, Inc. SPARCstation IPC workstation. This is a RISC based workstation with a 16 inch color monitor, 12 MByte RAM and a 207 Mbyte internal SCSI disk drive. As the reader will recognize, the administrator processor software could run on other hardware.

Detailed Description Text - DETX (59):

The administrator processor software preferably runs under a version of the UNIX operating system available from Sun Micro Systems, SunOS 4.1.1. UNIX is a portable multiuser operating system originally developed by AT&T Bell Labs. A relational database 85 is used to store data on the configuration of the message processors and the customers they serve. This database is also used to hold the statistical data retrieved from the message processors. The operating system under which the system runs should permit the use of a relational database system which has a number of advantages including a powerful, flexible high level language for producing reports and a user friendly MMI 87. The database program used in the preferred embodiment of the administrator processor is SYBASE with its associated Report Workbench 89, and APT-EDIT 91 forms and menus tool although the reader will recognize other database systems could be used. APT-EDIT 91 functions as a forms generator shown in FIG. 19 as Forms Generator 709.

Detailed Description Text - DETX (61):

Table 5, below, shows the hierarchy of forms in the administrator processor. At a highest level, a main window is displayed when a user logs on and the application is first accessed. The main window shows a choice of four options and the user selects one depending on the task to be performed. As three levels of additional security are provided through the database system, the user is prevented from seeing fields or forms he is not authorized to use. Any appropriate number of levels of security can be employed.

Detailed Description Text - DETX (68):

When the user completes entering (or updating) data on a database update form, the data is entered into the appropriate relation in the database. In Table 5 below, the Action column specifies the name of the main relation (in italics) that is to be updated in any of the Table Maintenance option forms. Table 6, below shows the administrator processor database relations.

Detailed Description Text - DETX (69):

After changing message processor data, if the updated tables are to be sent to message processors affected by the change, the user can either wait until the next scheduled transfer of statistics or can cause the update to be transmitted immediately by using the Immediate Message Processor Connection form. This causes activation of the APIF process to collect data and transfer it to the indicated message processors.

Detailed Description Text - DETX (70):

The APIF uses a file transfer protocol, such as Kermit to transfer data stored in ASCII files, for example, the message processor table updates to the message processor. The APIF uses a commercially available command language tool, for example, Tool Command Language and EXPECT, created by the National Institute of Standards and Technology to communicate with a shell process on the message processor to invoke various programs, such as, the message processor Data Loader that use the data to update the message processor.

Detailed Description Text - DETX (71):

When it is necessary to audit the message processor to ensure that its tables match the data in the administrator processor's database, the APIF calls the message processor, logs on and uses the message processor data unloader to unload the data from message processor tables and stores it on the message processor's disk in ASCII format. When the message processor data unloader exits, the APIF uses the file transfer protocol, such as Kermit to retrieve the file. Once the file has been retrieved, the administrator processor compares it against an message processor configuration file prepared from the current database.

Detailed Description Text - DETX (73):

1) Use the message processor Data Unloader program to get all transaction statistics records for each customer x (where x is a customer name from a Customer relation in the administrator processor database) and place the records into a results file. This is repeated for all customers on the message processor.

Detailed Description Text - DETX (77):

1) The file is separated into three parts in file separator 711. The transaction records (from the MP's Customer Transfer Statistics file as shown in Table 3(a)), an overall message processor statistics record (from the message processor statistics file as shown in Table 3(b)) and a set of customer MDR storage statistics (from the Customer Storage Statistics file as shown in Table 3(c)).

Detailed Description Text - DETX (79):

3) The Customer Transfer Statistics and Customer Storage Statistics records are inserted into the Message Transaction Relation and the Customer Storage relation respectively, using the SYBASE bulk load utility.

Detailed Description Text - DETX (81):

The statistics will subsequently be used in the reports produced by the Transaction Reports form. The Transaction Reports accessed from the Reports menu are predefined reports. They can be produced by using a commercially available database system, for example, by using the SYBASE REPORT WORKBENCH using SQL commands. For example, a Transaction Summary Report is produced using the following SQL statement:

Detailed Description Text - DETX (86):

This uses the MDR storage statistics (in the Customer Storage Relation) with the statistics on MDRs transferred to the customer (in the Message Transaction Relation) for the basic information. The Message Processor Table Relation and Customer Relations are joined with these to produce human readable names for the message processor and customer. The age of transactions of interest is determined by the report start date and report end date parameters which are taken from the values the user enters in the Transaction Summary Report form. A user my be given authority to create a custom report using SQL, but would not typically be given the ability to modify the standard reports.

Detailed Description Text - DETX (87):

Alarm messages are sent to the administrator processor for immediate attention by the System Administrator, transmission to a surveillance Operations System, printing on a dedicated alarm printer and/or logging for later review. Alarms may be generated within the administrator processor or sent from a message processor using a file transfer protocol such as Kermit. The alarm message will be in the Alarm Message File in the format shown in Table 2 and will contain full information about the alarm or alarms, if several have accumulated. This data may be entered into the database in various ways. The information should also be entered into the System Errors Relation. A message, an alert, is then sent to the Alarm Display process 93, for example, as implemented in the Alarm Transmitter 705, so that it alerts the System Administrator to the new alarm. The alert uses a colored alarm icon if the System Administrator is using a graphics workstation or a text message (when the current form is exited) if an ASCII terminal is being used.

Detailed Description Text - DETX (88):

The collector typically manages the collection of call detail records from multiple message processors and Customer's PBX SMDR storage devices and store them in a file for each customer. Thus, it separates the MDRs by customer and transmits the records to Customer provided call-accounting systems for costing calls and producing management reports. The collector may process the MDRs itself and send costed MDR and/or the management reports to the customer. The collector advantageously has a Customer relational database with the appropriate information to maintain the operational, administrative, provisioning and billing of collector services as well as a transaction log of all Customer transactions.

Detailed Description Text - DETX (89):

The collector should have some or all of the following capabilities:

Detailed Description Text - DETX (90):

Collect MDRs from message processors;

Detailed Description Text - DETX (91):

Collect SMDR from PBXs or collection devices;

Detailed Description Text - DETX (92):

Provide a file storage system for MDR (Data Element Standard Storage File, DESS File);

Detailed Description Text - DETX (93):

Provide the ability to search, display and report on MDR in the Data Element Standard Storage File through a secure, flexible report writer;

Detailed Description Text - DETX (96):

Provide the ability to rate MDRs in a real time environment to support the hospitality industry; Provide support for other applications to co-reside on the same platform or network and retrieve information from the DESS file;

Detailed Description Text - DETX (97):

Provide a secure method to access the collector through dial-up and network logins;

Detailed Description Text - DETX (99):

Report on consolidated network charges rolled-up according to the customer's organization structure;

Detailed Description Text - DETX (101):

Report and transmit errors and failures with data transfer, call detail collection, user transactions, system hardware, system power and system network, for integration to the telephone company.

Detailed Description Text - DETX (102):

The collector is advantageously based on a UNIX workstation. The preferred hardware embodiment is the Sun Microsystems, Inc. SPARCSTATION IPC workstation. (This is a RISC based workstation with a 16 inch color monitor, 8 MByte RAM and a 207 Mbyte internal SCSI disk drive.) The collector software could run on other hardware supporting the UNIX system.

Detailed Description Text - DETX (103):

The collector's application software runs under the UNIX operating system. UNIX is a portable multiuser operating system originally developed by AT&T Bell Labs. It is widely available on virtually every type of computer hardware allowing the collector's application software to be readily ported to another hardware platform. It is to be understood that the current invention comprises software running on other hardware platforms.

Detailed Description Text - DETX (104):

The collector application software is designed as a set of processes each of which carries out a specific function. It borrows portions of its design from that of the administrator processor including such capabilities as sending MDR to a customer's manager.

Detailed Description Text - DETX (105):

In addition, there should be other modules having other capabilities such as the MDR storage unit file (the DESS file) which acts as a server for all INFO MDR requests for call details. The DESS file is also the primary storage of all polled information from message processors, MDR from CPE systems and call records of manager sessions. The stored information is in flat files indexed by Company, Customer Name (Business ID), source location (message processor site) and date. Each file will maintain all of the call records for that day. The start time of the call will determine which day the record is stored in.

Detailed Description Text - DETX (106):

All information concerning the customer and the collector system is kept in a relational database called the Collector Customer Database.

Detailed Description Text - DETX (107):

We will now describe the steps involved in collecting MDR from a message processor, allocating them to files for each customer and then delivering them to the customer's manager. The collector polls message processors (step 2001) using Kermit and stores the files in the same format used by the message processor (the Bellcore AMA format (BAF) or the Ameritech Call Detail Recording System format (ACDRS) (step 2003). The collector will manage the transaction, initiate the poll and log the transaction statistics, such as number of call records transferred, into a relational database. This transaction information will be available to a flexible report writer for customer billing & statistical reports.

Detailed Description Text - DETX (108):

MDRS are rated (or the cost for a call calculated) by a Call Rating process (steps 2007, 2009). It provides analysis of traffic and cost information independent of the reporting structure necessary for Call Accounting. Cost and traffic analysis can be used to optimize carrier service selections and trunk loads for future capital investment. It may also be used to quickly determine unusual calling patterns and report on misuse, abuse or fraud. A call process rating receives DESS file call records, rates them and then stores the rated records in a separate relational database file (step 2011) which is to be used for subsequent analysis and reporting. It supports North American V & H rating, International V & H rating, International Meter Pulse rating and International Code Book rating for calls regardless of their point of origin or termination in the phone network.

Detailed Description Text - DETX (109):

The MDRs are delivered when the customer's manager polls the collector (step 2013).

Detailed Description Text - DETX (110):

The MDR can also be delivered in real-time and may also be rated prior to delivery to the customer. In this case, the call rating process receives individual call records from the DESS file rather than an entire file (step 2009) prior to rating and then passes each rated record to the real time delivery module (step 2015).

Detailed Description Text - DETX (111):

When the customer subscribes to the management reporting service, Management reports are produced. These are produced using the SYBASE flexible report writer with standard report procedures for analysis and/or summary reporting of the rated calls database. The reports are delivered to customer systems, for example General Ledge and Property Management systems through the available communication links. They can also be printed on the collector's printers and mailed to customers.

Detailed Description Text - DETX (112):

The collector my also collect statistics for use in producing management reports for the telephone company.

Detailed Description Text - DETX (113):

The collector sends reports, alarm messages or provisioning messages to a telephone company Operations Systems. This may advantageously be done using a Bellcore defined protocol carried on the LAN or RJE/SNA links.

Detailed Description Text - DETX (114):

The collector may request and receive MDRs from message processor and separately store them by customer.

Detailed Description Text - DETX (115):

Also, the collector may request and receive MDRs from MDR capture devices and separately store the MDRs by switch.

Detailed Description Text - DETX (116):

The collector can retrieve MDRs from capture devices using various protocols. Normally, collection begins when the collector initiates a poll (although some capture devices may have the capability of calling the collector when the unit reaches a certain storage threshold). The collector supports several models of capture devices 17 (including but not limited to devices manufactured by MOSCOM).

Detailed Description Text - DETX (117):

Collection of MDRs begins when the collector calls the remote PCSU. The MDRs are converted into the format selected for the customer before being stored in the DESS file.

Detailed Description Text - DETX (118):

The collector supports costing of MDRs and delivery of MDRs to customers.

Detailed Description Text - DETX (119):

Managers (or other call accounting systems) may login to the collector using standard ASCII terminals or PC terminal emulators and can retrieve MDRs from the DESS database using the protocol, for example, Kermit.

Detailed Description Text - DETX (120):

The collector also supports management report production and management report delivery.

Detailed Description Text - DETX (121):

The manager collects MDRs for a single customer from multiple message processors, collectors and simplified MDR capture devices and prepares various management reports such as organization reports to allocate network charges to departments within an organization, traffic and call distribution reports used for network planning, facility reports on all calls terminating/originating/ transiting a network node, detailed billing reports by extension number of account code, and other user definable reports. MOSCOM's MaxNet and Discovery/1 products can be used as managers. Several other call accounting systems that can be used as managers also exist. In order for them to work with this invention, they merely need accept MDR in any one of the output formats produced by the collector or message processor.

Detailed Description Paragraph Table - DETL (2):

TABLE 1

Message Processor Processes Process Name Element Description

Installation Process A1 The Installation Process starts the processes necessary to allow the AP to download MP executable software and data to the MP and disk. Initialization Process A2 The Initialization Process is responsible for starting certain MP processes that run continuously (e.g. Switch Interface). Database Update Notifier A3 The Database Update Notifier is responsible for notifying processes that a database update has occurred. MDR File Manager A4 The MDR File Manager is responsible for initializing the MDR file system and for adding and removing Customer MDR Files. Switch Interface A5 The Switch Interface receives MDRs from a telephone switch, translates them to the customer's MDR format, and stores them in the appropriate Customer MDR file. Statistics Process A6 The

Statistics Process is responsible for the periodic gathering of MDR file statistics and the initialization/termination of records in the MP Statistics File and the Customer Statistics Files. Communications Configurer A7 The Communications Configurer is responsible for configuring the various ports used by Customer Premises Interfaces, setting up customer log-ins on the MP, and for executing Customer Premises Interfaces on the appropriate ports. Single Direct Customer A8 The Single Direct Customer Premises Premises Interface Interface immediately transmits MDRs to a single customer. Multiple Direct Customer A9 The Multiple Direct Customer Premises Premises Interface Interface immediately transmits MDRs to multiple customers. Customer Premises Scheduler A10 The Customer Premises Interface executes Interface Scheduler the Scheduled Customer Premises Interface at previously scheduled times. Scheduled Customer A11 The Customer Premises Interface transmits Premises Interface MDRs to customer premises at previously scheduled intervals. Polled Customer Premises A12 The Customer Premises Interface sends Interface MDRs to the customer at the customer's initiation. Kermit File Builder A13 The Kermit File Builder reads MDRs from Customer MDR Files and places them in Discovery Files for Kermit to transfer to a Discovery/Centrex product. Alarm Interface A14 The Alarm Interface sends alarm messages to the alarm port on the AP. Kermit Login Process A17 The Kermit Login Process is responsible for allowing login using kermit protocol. AP Interface A18 The AP Interface is responsible for interfacing to the Administrator Processor. Errored MDR Reprocessor (not The Errored MDR Reprocessor reads MDRs shown) from the Errored MDR File and attempts to determine the Customer MDR File in which to write them. Overflow MDR Reprocessor (not The Overflow MDR Reprocessor reads shown) MDRs from the Overflow MDR File for a customer and attempts to write them to the Customer MDR File for the customer. MP Data Loader (not The MP Data Loader is responsible for shown)

loading formatted ASCII data into the MP data files. MP Data Unloader (not The MP Data Unloader is responsible for shown) converting data from MP data files into formatted ASCII data. MP Data Entry Process (not The MP Data Entry Process is an interactive shown) process that formats MP data file contents for viewing and allows entry of data into MP data files on an item-by-item basis. MP Pipe Message Generator (not The MP Pipe Message Generator is a process shown) that creates and sends pipe messages to processes. Its primary purpose is to create pseudo messages for testing.

Detailed Description Paragraph Table - DETL (4):

TABLE 7(b) ____________________________________ Bellcore AMA Format Modules Module Classi- Code fication Description ____________________________________ 0 AMA Terminating Module Code 20 AMA Carrier Access Terminating 21 AMA Carrier Access Originating 22 AMA Long Duration Call 23 AMA WATS 24 AMA PSDS 25 AMA Circuit Seizure/Release 900-999 AMA Reserved for BOC Use 70 ISDN Core Module 71 ISDN Abbreviated Module 72 ISDN Daily Aggregated Service Event 73 ISDN Basic Business Group 74 ISDN EKTS Capabilities 75 ISDN Call Pickup 76 ISDN Terminating Access Service 77 ISDN Terminating User Service 78 ISDN Calling Number Identification 79 ISDN Early Cut Through 101 MDR Digits Dialed 102 MDR Authorization Code 103 MDR Account Code 105 MDR Message Detail Recording 106 MDR Facility Identification 107 MDR Business Features

____

Detailed Description Paragraph Table - DETL (10):

TABLE 3(b) ____________________________________ MP Statistics File Format A B Field Contents ____________________________________ mps.sub.-- start start time for

transaction period mps.sub.-- end stop time for transaction period mps.sub.-- sw.sub.-- trans total MDR's read from switch mps.sub.-- err.sub.-- cnt number of errored MDR's read from switch mps.sub.-- filter-count total MDR's ignored mps.sub.-- err.sub.-- time time last errored MDR was read mps.sub.-- bad.sub.-- cust number of MDR's with a bad customer id mps.sub.-- bad.sub.-- size number of MDR's with wrong length mps.sub.-- bad.sub.-- fld number of MDR's with bad data fields mps.sub.-- mdr.sub.-- stor total MDR's stored in files mps.sub.-- mdr.sub.-- size MDR disk usage mps.sub.-- mp.sub.-- stat MP operational status mps.sub.-- overflow.sub.-- count total MDR's stored in the Overflow MDR File

____

Detailed Description Paragraph Table - DETL (11):

TABLE 3(c) ______________________________________ Customer Storage Statistics File Format A B Field Contents ______________________________________ css.sub.-- cust.sub.-- index identifies the customer css.sub.-- start.sub.-- line start time for statistics period css.sub.-- end-time stop time for statistics period css.sub.-- sw.sub.-- mdrs total MDR's read from switch for customer css.sub.-- dks.sub.-- mdrs total MDR's written to disk for customer css.sub.-- max-fsize maximum Cust. MDR File size reached ______________________________________

Detailed Description Paragraph Table - DETL (12):

TABLE 4 ______________________________________ Administrator Processor Software Modules Module Name Description ______________________________________ Admin.sub.-- proc The main AP process. The user interface will be a part of this section along with additional scripts to schedule reports, view logs, handle report output, etc. Reports The various reports required for this system will be created in the Report.sub.-- Workbench, stored

and executed from the Report.sub.-- Scheduler process. Report- This process accesses the database to determine Scheduler which reports have recently been run and need to have their scheduling data updated. Note that the initial cron entry for a scheduled report will be taken care of by the report scheduling form. Transaction.sub.-- This process will access the appropriate MP at Scheduler the times given in the transfer schedule table to accept transaction data and load it into the database. Time.sub.-- Keeper This process will send messages to the Transaction.sub.-- Scheduler to cause it to check for scheduled operations. Transaction.sub.-- This process will execute on a daily basis to Purger remove the transaction records that have exceeded their allowed time in the database. APIF This process will communicate with designated MPs to retrieve statistics, load software, and load/unload MP tables. Alarms from MPs will also go through this process. Alarm.sub.-- This process produces a real time notification of Display alarm messages on the screen if required. ______________________________________

Detailed Description Paragraph Table - DETL (13):

TABLE 5

AP Forms Hierarchy Option Form Selection Sec Windows Action Description

---

Table System Access Customer Form Update DB Defines MP and Customers for a Maintenance Customer particular company Customer Transfer Update DB Defines message transfer schedules Schedule Form Transfer.sub.-- Schedule for customers polled by the MP MP Company Form Update DB Stores information on a particular Customer company AP User Form Update DB Defines the users that can access the AP User AP including login ID, password and permissions System AP Destinations Update DB Defines destinations for sending

Configuration Form AP Destinations reports AP Parameters Form Update DB Defines AP parameters such as the AP Table number of days to store historical data, dedicated alarm port, dial-up number, baud rate AP Ports Form Update DB Defines each port and its parameters AP Ports for communicating with MPs MP Parameters Update DB Defines MP specific parameters (e.g. MP Table MP name, loaction, message storage capacity) MP Switch Interface Update DB Defines MP port used to connect MP Ports to switch MP-AP Interface Update DB Defines MP port used to connect to AP MP Alarm Interface Update DB Defines MP port used for delivery of alarms MP Ports Update DB Defines MP ports used to deliver MDR Transaction Process Call Reprocess Overflow Appl Function Requests reprocessing of overflowed Processing Information MDR MDR on a specific MP Form Delete Overflow Appl Function Requests the deletion of overflowed MDRs MDR on a specific MP Reprocess Error Appl Function Requests reprocessing of error MDR MDR on a specific MP Reprocess Error Appl Function Requests the deletion of error MDR MDR on a specific MP Immediate MP Appl Function Requests immediate connection to Connection a MP to update tables. View Error Appl Function View errored MDR sent to Records AP Reports Table Reports System Access Customer Report DB Report Report on all customers Reports Customer DB Report Reports the transfer schedule for Transfer each customer from every MP Schedule Report Company Report DB Report Report on all companies served by MPs AP User Report Reports on AP Users with userid and permissions System Configuration AP Destinations DB Report Reports on AP report destinations Reports Report AP Parameters DB Report Report on AP Parameters Report AP Ports Report DB Report Lists AP ports and their configurations MP Parameters DB Report Reports on MP parameters (MP Report name, Location, disk capacity, switch type and format, thresholds MP Ports Report DB Report Reports on MP parameters (MP name, Location, disk capacity, switch

type and format, thresholds MP Audit Report Application Report Report on discrepancies between AP database and tables stored in MPs. Transaction Transaction Detail DB Report Same as Transaction Summary Re- Reports Report port but also shows length of each transaction Transaction Summary DB Report Summarizes the number of records Report transferred and stored by MP and customer for a date range. Message Throughput DB Report Reports statistics on transfers between Report switches and MPs as well as between MPs and customer premises. Disk Capacity Detail DB Report Reports capacity and storage of Report MDRs for companies and MPs. Disk Capacity DB Report Same as Call Record Capacity Detail Summary Report Report except summarized by MP. Error Reports System Alarm Report DB Report Reports alarms that have occurred for specific MPs and time ranges. MDR Error Report DB Report Reports errors that would prevent processing of MDR by specific MPs. System Report List Schedule Report DB Report Lists currently scheduled reports for Maintenance Scheduler a given use View Log Files Appl Function View application log files for trouble- shooting purposes DB Backup or Backup AP User Database Function Backs up database of user information Recovery Database and system tables. Backup Application Database Function Backs up AP application relations Database Backup Transaction Database Function Backs up just SYBASE transaction Log logs Clear Transaction Database Function Clears out SYBASE transaction logs. Log Restore Database Database Function Instructions on how to restore the AP application database. Restore Transaction Database Function Instructions on how to reprocess Log transactions to bring database up to date.

Detailed Description Paragraph Table - DETL (14):

TABLE 6 ______________________________ AP Database

Relations Type/ Field Name Length Description

MP Table Relation This table is the primary source of information on an MP. mp.sub.-- index int MP index (1-500) mp.sub.-- name char(20) name of the MP mp.sub.-- loc char(20) MP location mp.sub.-- fname char(10) MP contact first name mp.sub.-- lname char(20) MP contact last name mp.sub.-- phone char(20) MP contact phone mp.sub.-- timezone int MP timezone - like Unix mp.sub.-- dst char(1) Daylight Savings Time used mp.sub.-- bill.sub.-- day int Reserved for future use mp.sub.-- sw.sub.-- fmt char(8) MP switch format mp.sub.-- sw.sub.-- prot char(8) MP switch protocol mp.sub.-- upd int AP updates MP - day bit map mp.sub.-- up.sub.-- time int AP updates MP - hour mp.sub.-- mdr.sub.-- cap int MDR cap in MB mp.sub.-- ovfl.sub.-- cap int overflow capacity in MB mp.sub.-- err.sub.-- cap int error file capacity in MB mp.sub.-- err.sub.-- thresh int error threshold count mp.sub.-- err.sub.-- pe int threshold count period (minutes) mp.sub.-- ap.sub.-- con.sub.-- typ char(1) AP connection type (D/P) mp.sub.-- alarm.sub.-- prefix char(30) prefix for alarm calls Message Transaction Relation This relation stores statistics on transfers from the MP to Manager cts.sub.-- mp.sub.-- index int MP identifier (1-500) cts.sub.-- cust.sub.-- index int cust id (1-500) cts.sub.-- start datetime start stats period cts.sub.-- end datetime end stats period cts.sub.-- tran.sub.-- type char(1) transfer type - D,S or P cts.sub.-- mdrs.sub.-- sent int MDR sent to CP Customer Storage Relation This relation stores statistics relating to MDR storage for a customer during a statistics collection period. css.sub.-- mp.sub.-- index int MP identifier (1-500) css.sub.-- cust.sub.-- index int cust id (1-500) css.sub.-- start datetime start stats period css.sub.-- end datetime end stats period css.sub.-- sw.sub.-- mdrs int MDRs read for cust css.sub.-- dsk.sub.-- mdrs int MDRs written to disk css.sub.-- max.sub.-- fsize int max MDR file size for cust MP Statistics Relation This relation stores statistics on the overall MP. mps.sub.-- mp.sub.-- index int MP identifier mps.sub.-- start datetime start of trans period mps.sub.-- end datetime end of trans period mps.sub.– sw.sub.– trans int MDR read from switch mps.sub.– err.sub.– cnt int errored MDRs read in mps.sub.– filter.sub.– count int records filtered out mps.sub.– err.sub.– time datetime time of last errored MDR mps.sub.– bad.sub.– cust int no. MDRs with bad cust id mps.sub.– bad.sub.– size int no. MDRs with bad length mps.sub.– bad.sub.– fld int no. MDRs with bad data flds mps.sub.– mdr.sub.– stor int tot MDRs stored in files mps.sub.– mdr.sub.– size int MDR disk usage mps.sub.– mp.sub.– stat int MP status mps.sub.– overflow.sub.– count int number recs in overflow area System Errors Relation This relation stores information on error messages. err.sub.– mp.sub.– index int MP identifier err.sub.– date.sub.– time datetime time of error occ. err.sub.– pid int process id of proc with error err.sub.– proc.sub.– type int type of errored process err.sub.– severity int error severity err.sub.– code int error id code err.sub.– mdrs int no. unprocessed MDRs err.sub.– desc char(255) desc of error from MP err.sub.– type char(1) alarm (A) or error msg (E) Customer Relation This relation stores information on a customer. cust.sub.– index int cust in MP (1-500) cust.sub.– comp.sub.– index int cust's company cust.sub.– bci char(20) business cust id (MDR ID) cust.sub.– bci.sub.– 2 char (20) second alphanumeric bci field cust.sub.– name char(20) customer name cust.sub.– fname char(10) cust contact first name cust.sub.– lname char(20) cust contact last name cust.sub.– phone char(20) contact phone number cust.sub.– mp.sub.– index int MP used by customer (1-500) cust.sub.– dat.sub.– tran char(1) data transfer method (P,D,S) cust.sub.– port.sub.– grp char(20) grp used to contact CP cust.sub.– tran.sub.– fmt char(10) transfer format cust.sub.– port.sub.– name char(20) NOT USED -name of port to use cust.sub.– protocol char(8) transfer protocol cust.sub.– cp.sub.– phone char (30) CP phone to connect to CA prdt cust.sub.– cp.sub.– user char(20) CP login cust.sub.– passwd char(20) CP login password cust.sub.– parity char(1) CP parity cust.sub.– baud int CP baud rate cust.sub.-- stop int CP stop bits cust.sub.-- data int CP data bits cust.sub.-- flow char(1) CP flow control flag (Y/N) cust.sub.-- mp.sub.-- enabled char(1) MP enabled flag for cust cust.sub.-- mp.sub.-- ovrfl char(1) MP handles overflow flag cust.sub.-- mp.sub.-- disk int MP disk alloc for cust MDRs cust.sub.-- oth.sub.-- disk int NOT USED -MP alloc for other files Transfer Schedule Relation This relation stores the schedule for transfers to the customer's Manager. trns.sub.-- index int unique transaction id number trns.sub.-- cust.sub.-- index int customer index (1-500) trns.sub.-- mp.sub.-- index int MP index (1-500) trns.sub.-- day int day bit map trns.sub.-- time int hour of transfer

---

Claims Text - CLTX (1):

1. A universal MDR data record collection and reporting system for use with Centrex switches on a telephone network, with a message processor comprising:

Claims Text - CLTX (2):

(a) a receiver, said receiver receiving an MDR in one of a plurality of MDR formats from a one of a plurality of different central office switches of a telephone network;

Claims Text - CLTX (4):

(c) means for storing the reformatted MDR in an MDR file; and

Claims Text - CLTX (6):

2. The system of claim 1, further comprising a statistics collector collecting and storing statistics from a plurality of different central office switches.

Claims Text - CLTX (7):

3. The system of claim 1, further comprising storage storing the selected format and the MDR format for a type of switch.

Claims Text - CLTX (8):

4. The system of claim 1, further comprising an alarm collector collecting and simultaneously storing message processor alarms from a plurality of different central office switches.

Claims Text - CLTX (9):

5. The system of claim 1, wherein there is at least one MDR file for each customer simultaneously using a plurality of different central office switches on the telephone network.

Claims Text - CLTX (10):

6. The system of claim 1, further comprising an error detector determining if the MDR is in the format for a central office switch, and an error file storing a plurality of MDRs not in the format for the switch.

Claims Text - CLTX (21):

12. The system of claim 8, further comprises storage means for storing selected formats for a plurality of customers.

Claims Text - CLTX (22):

13. The system of claim 8, further comprising a third transmitter for collecting and storing a plurality of administrator processor alarms.

Claims Text - CLTX (30):

(c) storage storing the reformatted MDRs.

Claims Text - CLTX (31):

19. The system of claim 18, further comprising a collector, including:

Claims Text - CLTX (36):

(e) storage storing the reformatted MDRs in the customer files; and

Claims Text - CLTX (37):

(f) an allocator allocating and storing the MDRs by switch.

Claims Text - CLTX (38):

20. A universal method for collecting and reporting MDR data records for use with Centrex switches on a telephone network, comprising the steps of:

Claims Text - CLTX (39):

(a) receiving an MDR in one of a plurality of MDR formats from a one of a plurality of different central office switches a telephone network;

Claims Text - CLTX (41):

(c) storing the reformatted MDR in an MDR file in a message; and

Claims Text - CLTX (43):

21. The method of claim 20, further comprising collecting and storing statistics from a plurality of different central office switches.

Claims Text - CLTX (44):

22. The method of claim 20, further comprising storing the

selected format and the MDR format for that type of switch.

Claims Text - CLTX (45):

23. The method of claim 20, further comprising collecting and simultaneously storing message processor alarms from a plurality of different central office switches.

Claims Text - CLTX (46):

24. The method of claim 20, further comprising simultaneously storing MDRs separately for each customer using a plurality of different central office switches on the telephone network.

Claims Text - CLTX (47):

25. The method of claim 20, further comprising determining if the MDR is in the format for a central office switch switch, and storing a plurality of MDRs not in the format for the switch.

Claims Text - CLTX (54):

(d) transmitting and storing the selected format; and

Claims Text - CLTX (55):

(e) transmitting and storing the MDR format.

Claims Text - CLTX (59):

32. The method of claim 20, further comprising storing selected formats for a plurality of customers.

Claims Text - CLTX (60):

33. The method of claim 20, further comprising collecting and storing a plurality of administrator processor alarms.

Claims Text - CLTX (66):

(b) receiving and storing reformatted MDRs in the manager.

Claims Text - CLTX (68):

(a) communicating between a collector and the message processor;

Claims Text - CLTX (69):

(b) communicating between the collector and the manager;

Claims Text - CLTX (70):

(c) receiving reformatted MDRs in the collector;

Claims Text - CLTX (71):

(d) allocating and storing MDRs by customer in the collector; and

Claims Text - CLTX (72):

(e) allocating and storing MDRs by switch in the collector.